
**Oblika izmenjave tehničnih podatkov za uporabo v industrijskem inženiringu
avtomatizacije sistemov - Označevalni jezik za avtomatizacijo (AvtomatizacijaML) -
4. del: Logika**

Engineering data exchange format for use in industrial automation systems engineering
- Automation Markup Language - Part 4: Logic

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[SIST EN IEC 62714-4:2020](https://standards.iteh.ai/catalog/standards/sist/285bae7a-9ae4-407e-a0f3-856b380ba734/sist-en-62714-4-2019)

<https://standards.iteh.ai/catalog/standards/sist/285bae7a-9ae4-407e-a0f3-856b380ba734/sist-en-62714-4-2019>

Ta slovenski standard je istoveten z: prEN IEC 62714-4:2019

ICS:

25.040.40	Merjenje in krmiljenje industrijskih postopkov	Industrial process measurement and control
35.060	Jeziki, ki se uporabljajo v informacijski tehniki in tehnologiji	Languages used in information technology
35.240.50	Uporabniške rešitve IT v industriji	IT applications in industry

oSIST prEN IEC 62714-4:2019

en,fr,de



65E/654/CDV

COMMITTEE DRAFT FOR VOTE (CDV)

PROJECT NUMBER: IEC 62714-4 ED1	
DATE OF CIRCULATION: 2019-05-17	CLOSING DATE FOR VOTING: 2019-08-09
SUPERSEDES DOCUMENTS: 65E/619/CD, 65E/651/CC	

IEC SC 65E : DEVICES AND INTEGRATION IN ENTERPRISE SYSTEMS	
SECRETARIAT: United States of America	SECRETARY: Mr Donald (Bob) Lattimer
OF INTEREST TO THE FOLLOWING COMMITTEES:	PROPOSED HORIZONTAL STANDARD: <input type="checkbox"/> Other TC/SCs are requested to indicate their interest, if any, in this CDV to the secretary.
FUNCTIONS CONCERNED: <input type="checkbox"/> EMC <input type="checkbox"/> ENVIRONMENT <input type="checkbox"/> QUALITY ASSURANCE <input type="checkbox"/> SAFETY	
<input checked="" type="checkbox"/> SUBMITTED FOR CENELEC PARALLEL VOTING Attention IEC-CENELEC parallel voting The attention of IEC National Committees, members of CENELEC, is drawn to the fact that this Committee Draft for Vote (CDV) is submitted for parallel voting. The CENELEC members are invited to vote through the CENELEC online voting system.	<input type="checkbox"/> NOT SUBMITTED FOR CENELEC PARALLEL VOTING

This document is still under study and subject to change. It should not be used for reference purposes.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

TITLE:

Engineering data exchange format for use in industrial automation systems engineering - Automation Markup Language - Part 4: Logic

PROPOSED STABILITY DATE: 2024

NOTE FROM TC/SC OFFICERS:

CONTENTS

1		
2		
3	FOREWORD.....	9
4	INTRODUCTION.....	11
5	1 Scope.....	13
6	2 Normative references	13
7	3 Terms, definitions and abbreviations	14
8	3.1 Terms and definitions	14
9	3.2 Abbreviations	15
10	4 Conformity.....	15
11	5 Overview	16
12	5.1 Logic information in production system engineering.....	16
13	5.2 Logic models in production system engineering.....	17
14	5.3 Storing logic models in AML logic XML	18
15	5.4 Referencing logic information	20
16	6 Logic models	21
17	6.1 General	21
18	6.2 Gantt charts	21
19	6.2.1 General	21
20	6.2.2 Graphical elements.....	21
21	6.2.3 Chart structure	21
22	6.2.4 Logic information	22
23	6.2.5 Logic information within Gantt chart.....	22
24	6.3 Activity-on-node networks	23
25	6.3.1 General	23
26	6.3.2 Graphical elements.....	23
27	6.3.3 Node structure.....	23
28	6.3.4 Network structure	23
29	6.3.5 Logic information	24
30	6.3.6 Logic information within activity-on-node networks	24
31	6.4 Timing diagrams.....	25
32	6.4.1 General	25
33	6.4.2 Graphical elements.....	25
34	6.4.3 Diagram structure	25
35	6.4.4 Logic information	26
36	6.4.5 Logic information within timing diagrams	26
37	6.5 Sequential function charts	27
38	6.6 Function block diagrams.....	27
39	6.7 Mathematical expression	27
40	7 AML logic XML schema description	27
41	7.1 Schema overview	27
42	7.1.1 Use of IEC 61131-10 schema	27
43	7.1.2 Schema versioning	29
44	7.2 Root element “AMLLogic	30
45	7.2.1 General	30
46	7.2.2 Attributes.....	30
47	7.2.3 Sub-element “WriterHeader”	30

48	7.2.4	Sub-element “Types”	30
49	7.2.5	Sub-element “Documentation”	31
50	7.3	Complex type “FunctionBlock”	31
51	7.3.1	General	31
52	7.3.2	Attributes	31
53	7.3.3	Sub-element “Parameters”	31
54	7.3.4	Sub-element “Vars”	31
55	7.3.5	Sub-element “MainBody”	31
56	7.3.6	Complex type “ParameterSet”	32
57	7.3.7	Complex type “VariableDecl”	32
58	7.4	Complex Type “IML”	33
59	7.4.1	General	33
60	7.4.2	Attributes	33
61	7.4.3	Sub-element “Resource”	33
62	7.4.4	Sub-element “TimeInformation”	34
63	7.4.5	Choice of “IML***” element	34
64	7.4.6	Complex type “IMLStep”	34
65	7.4.7	Complex type “IMLTransition”	35
66	7.4.8	Complex type “IMLSimultaneousDivergence”	35
67	7.4.9	Complex type “IMLSimultaneousConvergence”	36
68	7.5	Complex Type “MathematicalExpression”	36
69	7.5.1	General	36
70	7.5.2	Attributes	36
71	7.5.3	Sub-element “VariableMapping”	36
72	7.5.4	Sub-element “MathML”	37
73	7.6	Simple type “LogicModelTypeEnum”	37
74	7.7	Simple type “TimeUnion”	37
75	7.8	Simple type “TimeFormatEnum”	37
76	7.9	Simple type “UuidString”	37
77	8	Storing logic models	37
78	8.1	General	37
79	8.2	Storing Gantt charts in AML logic XML	38
80	8.2.1	Common rules	38
81	8.2.2	Storing the start of a Gantt chart	38
82	8.2.3	Storing bars	38
83	8.2.4	Storing arrows	39
84	8.2.5	Storing successor bars	40
85	8.2.6	Storing predecessor bars	41
86	8.3	Storing activity-on-node networks in AML logic XML	42
87	8.3.1	Common rules	42
88	8.3.2	Storing the start of an activity-on-node network	42
89	8.3.3	Storing nodes	43
90	8.3.4	Storing arrows	43
91	8.3.5	Storing successor nodes	44
92	8.3.6	Storing predecessor nodes	45
93	8.4	Storing timing diagrams in AML logic XML	46
94	8.4.1	Common rules	46
95	8.4.2	Storing the timeline of a timing diagram	46
96	8.4.3	Storing resources and resource states	47

97	8.4.4	Storing lifelines.....	47
98	8.4.5	Storing the time signal and the resource signal.....	49
99	8.5	Storing sequential function charts in AML logic XML.....	51
100	8.5.1	Common rules	51
101	8.5.2	Storing variables	51
102	8.6	Storing function block diagrams in AML logic XML.....	52
103	8.6.1	Common rules	52
104	8.6.2	Storing variables	52
105	8.7	Storing mathematical expressions in AML logic XML	52
106	8.7.1	Common rules	52
107	8.7.2	Storing variables	53
108	8.7.3	Storing variable mappings	54
109	8.7.4	Storing mathematical expressions	54
110	9	Meta information about AML logic XML writer tools.....	54
111	10	Extensions of AML classes for logic.....	56
112	10.1	General.....	56
113	10.2	AutomationMLLogicRoleClassLib	56
114	10.2.1	General	56
115	10.2.2	RoleClass InterlockingTargetGroup	57
116	10.2.3	RoleClass InterlockingSourceGroup	57
117	10.2.4	RoleClass LogicModelObject	58
118	10.3	AutomationMLLogicInterfaceClassLib	58
119	10.3.1	General	58
120	10.3.2	InterfaceClass LogicModelInterface	59
121	10.3.3	InterfaceClass SequencingLogicModelInterface	59
122	10.3.4	InterfaceClass BehaviourLogicModelInterface	59
123	10.3.5	InterfaceClass InterlockingLogicModelInterface	60
124	10.3.6	InterfaceClass LogicModelElementInterface	60
125	10.3.7	InterfaceClass VariableInterface	60
126	10.3.8	InterfaceClass InterlockingVariableInterface	61
127	10.4	AutomationMLPLCopenXMLInterfaceClassLib	61
128	10.4.1	General	61
129	10.4.2	InterfaceClass VariableInterface	61
130	10.5	AutomationMLInterfaceClassLib	62
131	10.5.1	General	62
132	10.5.2	InterfaceClass InterlockingConnector	62
133	10.5.3	InterfaceClass PLCopenXMLInterface	62
134	11	Referencing AML logic XML documents.....	62
135	11.1	General.....	62
136	11.2	Referencing logic information	62
137	12	Linking AML objects with interlocking information	63
138	12.1	General.....	63
139	12.2	Referencing interlocking information.....	63
140	Annex A (Informative)	Examples for storing logic models in AML logic XML	65
141	A.1	Example for storing Gantt charts	65
142	A.1.1	Storing of activities without predecessor and successor relation	65
143	A.1.2	Storing of an activity sequence	65
144	A.1.3	Storing of an activity sequence with divergences	67

145	A.1.4 Storing of an activity sequence with convergences	68
146	A.2 Example for storing activity-on-node networks.....	69
147	A.2.1 Storing of activities without predecessor and successor relation	69
148	A.2.2 Storing of an activity sequence	70
149	A.2.3 Storing of an activity sequence with divergences	71
150	A.2.4 Storing of an activity sequence with convergences	72
151	A.3 Example for storing timing diagrams.....	73
152	A.3.1 Example of storing internal signal	73
153	A.3.2 Example of storing external signal	74
154	A.3.3 Example of storing signal between two resource states flows	76
155	A.4 Example for storing sequential function charts.....	77
156	A.5 Example for storing function block diagrams.....	80
157	A.6 Example for storing mathematical expressions	82
158	Annex B (Informative) Examples for referencing logic information	85
159	B.1 General	85
160	B.2 Referencing logic information expressed as logic models	85
161	B.2.1 Referencing logic information stored in one FunctionBlock	85
162	B.2.2 Referencing logic information, which is composed of several	
163	FunctionBlocks	86
164	B.2.3 Referencing logic information, which is composed of several AML	
165	logic XML documents	86
166	B.3 Referencing logic information as a part of logic models	87
167	B.3.1 Referencing a variable.....	88
168	B.3.2 Referencing a logic element	88
169	B.4 Referencing logic information as a part of already referenced logic models	89
170	Annex C (Informative) Examples for referencing interlocking information	92
171	C.1 General	92
172	C.2 Interlocking information	93
173	C.3 Referencing interlocking information without interlocking condition	93
174	C.4 Referencing interlocking information with interlocking condition.....	95
175	Annex D (Normative) XML representation of AML standard libraries	99
176	D.1 AutomationMLLogicRoleClassLib	99
177	D.2 AutomationMLLogicInterfaceClassLib	100
178	D.3 AutomationMLPLCOpenXMLInterfaceClassLib	100
179	Annex E (Normative) XML representation of AML logic XML schema	102
180	Bibliography.....	106
181		
182	Figure 1 – Overview of the engineering data exchange format AML	11
183	Figure 2 – Example of system representation with roles of information in AML.....	17
184	Figure 3 – Logic models in AML	18
185	Figure 4 – Storing logic models in AML logic XML.....	19
186	Figure 5 – Modelling elements of the AML logic XML	20
187	Figure 6 – Model elements of Gantt charts.....	21
188	Figure 7 – Information provided by Gantt charts.....	22
189	Figure 8 – Model elements of activity-on-node networks	23
190	Figure 9 – Information provided by activity-on-node networks	25
191	Figure 10 – Model elements of timing diagrams	25

192	Figure 11 – Information provided by timing diagrams	27
193	Figure 12 – AML logic schema overview	29
194	Figure 13 – Root element “AMLogic”	30
195	Figure 14 – Complex type “FunctionBlock”	31
196	Figure 15 – Complex type “ParameterSet”	32
197	Figure 16 – Complex type “VariableDecl”	32
198	Figure 17 – Complex type “IML”	33
199	Figure 18 – Complex type “IMLStep”	34
200	Figure 19 – Complex type “IMLTransition”	35
201	Figure 20 – Complex type “IMLSimultaneousDivergence”	36
202	Figure 21 – Complex type “IMLSimultaneousConvergence”	36
203	Figure 22 – Complex type “MathematicalExpression”	36
204	Figure 23 – AutomationMLLogicRoleClassLib	56
205	Figure 24 – AutomationMLLogicInterfaceClassLib	58
206	Figure 25 – AutomationMLPLCopenXMLInterfaceClassLib	61
207	Figure A.1 – Flow rate of valves	82
208	Figure A.2 – Example for storing a mathematical expression	84
209	Figure B.1 – Referencing logic information (as SFC) stored in one FunctionBlock	85
210	Figure B.2 – XML text of the CAEX file for referencing logic information stored in one	
211	FunctionBlock	86
212	Figure B.3 – Referencing logic information, which is composed of several	
213	FunctionBlocks	86
214	Figure B.4 – Referencing logic information which is composed of several AML logic	
215	XML documents	87
216	Figure B.5 – XML text of the CAEX file for referencing logic information, which is	
217	composed of several AML logic XML documents	87
218	Figure B.6 – Referencing a variable	88
219	Figure B.7 – XML text of the CAEX file for referencing a variable	88
220	Figure B.8 – Referencing a logic element	89
221	Figure B.9 – XML text of the CAEX file for referencing a logic element	89
222	Figure B.10 – Referencing a variable of an already referenced logic model	90
223	Figure B.11 – XML text of the CAEX file for referencing a variable of an already	
224	referenced logic model	90
225	Figure C.1 – Example manufacturing system	92
226	Figure C.2 – Example interlocking source group and interlocking target group	93
227	Figure C.3 – Referencing interlocking information without interlocking condition	94
228	Figure C.4 – XML text of the CAEX file for referencing interlocking information without	
229	interlocking condition	95
230	Figure C.5 – Referencing interlocking information with interlocking condition	96
231	Figure C.6 – XML text of the CAEX file for referencing interlocking information with	
232	interlocking condition	97
233	Figure C.7 – Linking logical interface with physical interface (extension to Figure C.5)	98
234	Figure C.8 – XML text of the CAEX file for linking logical interface with physical	
235	interface (extension to Figure C.6)	98
236		
237	Table 1 – Abbreviations	15

238	Table 2 – Mapping the start of a Gantt chart to IML element	38
239	Table 3 – Mapping of Gantt chart bars to IML elements	39
240	Table 4 – Mapping of Gantt chart arrow to IML elements	39
241	Table 5 – Mapping of Gantt chart bar with one or more successor bars to IML elements	40
242	Table 6 – Mapping of Gantt chart bar with one and more predecessor bars to IML	
243	elements	41
244	Table 7 – Mapping of the start of an activity-on-node network to IML elements	43
245	Table 8 – Mapping of activity-on-node network nodes to IML elements	43
246	Table 9 – Mapping of activity-on-node network arrows to IML elements	43
247	Table 10 – Mapping of activity-on-node network successor nodes to IML elements	44
248	Table 11 – Mapping of activity-on-node network predecessor nodes to IML elements	45
249	Table 12 – Mapping of the timeline of a timing diagram to IML elements	47
250	Table 13 – Mapping of timing diagram resources and resource states to IML elements	47
251	Table 14 – Mapping of timing diagram lifelines to IML elements	48
252	Table 15 – Mapping of the time signal and the resource signal to IML elements	50
253	Table 16 – Storing variable of a sequential function chart in AML logic XML	51
254	Table 17 – Storing variable of a function block diagram in AML logic XML	52
255	Table 18 – Storing variable of a mathematical expression in AML logic XML	53
256	Table 19 – Storing variable mappings of a mathematical expression in AML logic XML	54
257	Table 20 – Storing a mathematical expression in AML logic XML	54
258	Table 21 – Meta information about each AML logic XML writer tool	55
259	Table 22 – RoleClass InterlockingTargetGroup	57
260	Table 23 – RoleClass InterlockingSourceGroup	57
261	Table 24 – RoleClass LogicModelObject	58
262	Table 25 – InterfaceClass LogicModelInterface	59
263	Table 26 – InterfaceClass SequencingLogicModelInterface	59
264	Table 27 – InterfaceClass BehaviourLogicModelInterface	59
265	Table 28 – InterfaceClass InterlockingLogicModelInterface	60
266	Table 29 – InterfaceClass LogicModelElementInterface	60
267	Table 30 – InterfaceClass VariableInterface	60
268	Table 31 – InterfaceClass InterlockingVariableInterface	61
269	Table 32 – InterfaceClass VariableInterface	61
270	Table 33 – InterfaceClass InterlockingConnector	62
271	Table 34 – InterfaceClass PLCopenXMLInterface	62
272	Table A.1 – Storing of the Gantt chart example “activities without predecessor and	
273	successor relations”	65
274	Table A.2 – Storing of the Gantt chart example “activity sequence”	66
275	Table A.3 – Storing of the Gantt chart example “activity sequence with divergence”	67
276	Table A.4 – Storing of the Gantt chart example “activity sequence with convergences”	68
277	Table A.5 – Storing of the activity-on-node network example “activities without	
278	predecessor and successor relations”	69
279	Table A.6 – Storing of the activity-on-node network example “activity sequence”	70
280	Table A.7 – Storing of the activity-on-node network example “activity sequence with	
281	divergence”	71

282	Table A.8 – Storing of the activity-on-node network example “activity sequence with	
283	convergences”	72
284	Table A.9 – Storing of the timing diagram example “transition from a state change to	
285	the subsequent state”	73
286	Table A.10 – Mapping of the timing diagram example “two external signals fired with	
287	delay of three seconds”	75
288	Table A.11 – Storing of the timing diagram example “signal fired by one resource state	
289	and consumed by another”	76
290	Table A.12 – Example for storing sequential function chart	78
291	Table A.13 – Example for storing a function block diagram	80
292		

iTeh STANDARD PREVIEW
(standards.iteh.ai)

SIST EN IEC 62714-4:2020

<https://standards.iteh.ai/catalog/standards/sist/285bae7a-9ae4-407e-a0f3-886b380ba734/sist-en-iec-62714-4-2020>

INTERNATIONAL ELECTROTECHNICAL COMMISSION

ENGINEERING DATA EXCHANGE FORMAT FOR USE IN INDUSTRIAL AUTOMATION SYSTEMS ENGINEERING – Automation Markup Language –

Part 4: Logic

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62714-4 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

The text of this standard is based on the following documents:

FDIS	Report on voting
65E/XXX/FDIS	65E/XX/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

345 A list of all parts in the IEC 62714 series, published under the general title *Engineering data*
346 *exchange format for use in industrial systems engineering – Automation Markup Language*,
347 can be found on the IEC website.

348 The committee has decided that the contents of this publication will remain unchanged until
349 the stability date indicated on the IEC web site under "http://webstore.iec.ch" in the data re-
350 lated to the specific publication. At this date, the publication will be

- 351 • reconfirmed,
352 • withdrawn,
353 • replaced by a revised edition, or
354 • amended.

355

356 The National Committees are requested to note that for this publication the stability date
357 is 2023.

358 THIS TEXT IS INCLUDED FOR THE INFORMATION OF THE NATIONAL COMMITTEES AND WILL BE
359 DELETED AT THE PUBLICATION STAGE.

360

361

iTeh STANDARD PREVIEW
(standards.iteh.ai)

SIST EN IEC 62714-4:2020

<https://standards.iteh.ai/catalog/standards/sist/285bae7a-9ae4-407e-a0f3-886b380ba734/sist-en-iec-62714-4-2020>

INTRODUCTION

The data exchange format defined in IEC 62714 (Automation Markup Language (AML)) is an XML schema based data format and has been developed in order to support the data exchange between engineering tools in a heterogeneous engineering tool landscape. IEC 62714-1 gives an overview about the format.

The goal of AML is to interconnect engineering tools from the existing heterogeneous tool landscape in their different disciplines, e.g. mechanical plant engineering, electrical design, process engineering, process control engineering, HMI development, PLC programming, robot programming, etc.

AML stores engineering information following the object oriented paradigm and allows modelling of physical and logical plant components as data objects encapsulating different aspects. An object may consist of other sub-objects and may itself be part of a larger composition or aggregation. Typical objects in plant automation comprise information on topology, geometry, kinematics, and logic, whereas logic comprises sequencing, behaviour, and control.

AML combines existing industry data formats that are designed for the storage and exchange of different aspects of engineering information. These data formats are used on “as-is” basis within their own specifications and are not branched for AML needs.

The core of AML is the top-level data format CAEX that connects the different data formats. Therefore, AML has an inherent distributed document architecture.

Figure 1 illustrates the basic AML architecture and the distribution of topology, geometry, kinematic, and logic information.

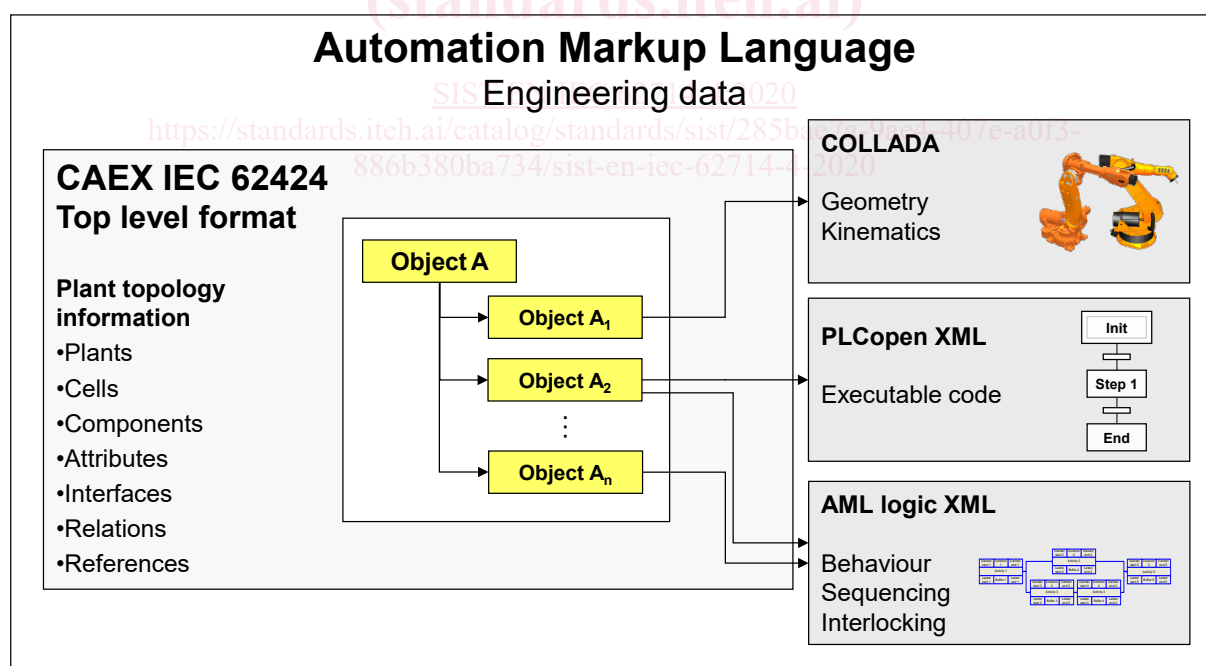


Figure 1 – Overview of the engineering data exchange format AML

387 Due to the different aspects of AML, IEC 62714 consists of different parts focussing on
 388 different aspects.

- 389 • IEC 62714-1: Architecture and general requirements

390 This part specifies the general AML architecture, the modelling of engineering data,
 391 classes, instances, relations, references, hierarchies, basic AML libraries and extended
 392 AML concepts.

- 393 • IEC 62714-2: Role class libraries

394 This part specifies additional AML libraries.

- 395 • IEC 62714-3: Geometry and kinematics

396 This part specifies the modelling of geometry and kinematics information.

- 397 • IEC 62714-4: Logic

398 This part specifies the modelling and referencing of logic information.

399 Further parts may be added in the future in order to interconnect further data standards to
 400 AML.

401 Clause 5 gives an informative overview of this part of the standard.

402 Clause 6 gives a normative description of the considered logic models.

403 Clause 7 gives a normative description of the AML logic XML schema, with which logic
 404 models can be stored.

405 Clause 8 specifies the normative provisions to store the logic models in AML logic XML.

406 Clause 9 defines how to store meta information about the source tool directly into the AML
 407 logic XML document.

408 Clause 10 defines a logic related role class library and interface class library.

409 Clause 10.4.2 gives a normative description regarding referencing logic information in AML
 410 logic XML documents.

411 Clause 12 gives a normative description regarding referencing interlocking information in AML
 412 logic XML documents.

413 Annex A provides examples for the storage of logic models in AML logic XML.

414 Annex B describes the referencing methods for logic information.

415 Annex C describes the referencing methods for interlocking information.

416 Annex D gives an informative XML representation of the libraries defined in this part of
 417 IEC 62714.

418 Annex E gives a normative XML representation of the AML logic XML schema defined in this
 419 part of IEC 62714.

420

ENGINEERING DATA EXCHANGE FORMAT FOR USE IN INDUSTRIAL AUTOMATION SYSTEMS ENGINEERING – Automation Markup Language –

Part 4: Logic

1 Scope

This part of IEC 62714 specifies the integration of logic information as part of an AML model for the data exchange in a heterogenous engineering tool landscape of production systems.

This part of IEC 62714 specifies three types of logic information: sequencing, behaviour, and interlocking information.

This part of IEC 62714 deals with the six following sequencing and behaviour logic models (covering the different phases of the engineering process of production systems) and how they are integrated in AML: Gantt chart, activity-on-node network, timing diagram, Sequential Function Chart (SFC), Function Block Diagram (FBD), and mathematical expression.

This part of IEC 62714 specifies how to model Gantt chart, activity-on-node network, and timing diagram and how they are stored in Intermediate Modelling Layer (IML).

NOTE 1 With this, it is possible to transform one logic model into another one. A forward transformation supports the information enrichment process and reduces or avoids a re-entry of information between the exchanging engineering tools.

NOTE 2 Mapping of other logic models, e.g. event-driven logic models like state charts, onto IML is possible.

This part of IEC 62714 specifies how interlocking information is modelled (as interlocking source and target groups) in AML. The interlocking logic model is stored in Function Block Diagram (FBD).

This part of IEC 62714 specifies the AML logic XML schema that stores the logic models by using IEC 61131-10.

This part of IEC 62714 specifies how to reference PLC programs stored in PLCopen XML documents.

This part of IEC 62714 does not define details of the data exchange procedure or implementation requirements for the import/export tools.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 62714-1:2014, *Engineering data exchange format for use in industrial automation systems engineering – Automation markup language – Part 1: Architecture and general requirements*

IEC 62424:2008, *Representation of process control engineering - Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools*

Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation 04 February 2004 (available at <<http://www.w3.org/TR/2004/REC-xml-20040204/>>) [viewed on 2016-07-28]

IEC 61131-3, *Programmable controllers - Part 3: Programming languages*